

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: 2001117769 A

(43) Date of publication of application: 27.04.01

(51) Int. Cl

G06F 9/06

(21) Application number: 11297759

(22) Date of filing: 20.10.99

(71) Applicant: MATSUSHITA ELECTRIC IND CO LTD

(72) Inventor: KANAMARU TOMOKAZU

(54) PROGRAM EXECUTING DEVICE

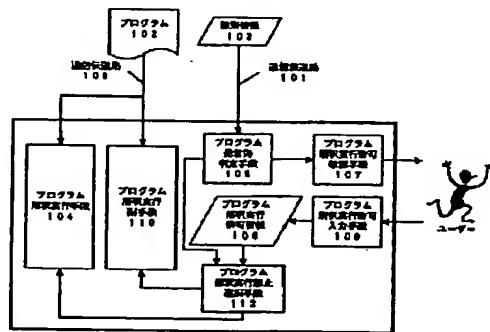
(57) Abstract:

PROBLEM TO BE SOLVED: To realize the assurance of safety, secrecy and completeness of a downloaded program conveniently at low cost in electrical household appliances having a network function for program distribution.

SOLUTION: This program executing device is constituted of a program interpretation executing means 104, a program interpretation executing sub-means 110, a program transmitting destination judging means 105, an execution permission of program interpretation input means 109 and an execution prohibition of program interpretation selecting means 112. When a judgment result of the program transmitting destination judging means 105 is affirmative, the execution prohibition of program interpretation selecting means 112 permits the program interpretation executing means 104 to interpret and execute a program 102 and when the judgment result is negative, the execution prohibition of program interpretation selecting

means 112 prohibits the program interpretation executing means 104 to interpret and execute the program 102.

COPYRIGHT: (C)2001,JPO



プログラム実行装置・その4

【特許請求の範囲】

【請求項1】 外部伝送路を通して送信される、プログラムと外部伝送路を通して送信される、プログラムの発信元の識別情報とを入力とする、プログラム実行装置であり、

プログラム実行装置は、

プログラムを解釈実行する手段である、プログラム解釈実行手段と、

識別情報が、信頼されたプログラム発信元を示すものであるか否かを判定する手段である、プログラム発信先判定手段と、

プログラム解釈実行禁止手段と、

からなり、

プログラム発信先判定手段の判定に従って、プログラム解釈実行禁止手段は、プログラム解釈実行手段がプログラムを解釈実行するのを禁止することを特徴とする、プログラム実行装置。

【請求項2】 外部伝送路を通して送信される、プログラムと外部伝送路を通して送信される、プログラムの発信元の識別情報とを入力とする、プログラム実行装置であり、

プログラム実行装置は、

前記プログラム解釈実行手段と、

前記プログラム発信先判定手段と、

前記プログラム解釈実行禁止手段と、

プログラムの解釈実行を許可する、あるいは、禁止する、のいずれかを示す、解釈実行許可情報を、ユーザが入力する手段である、プログラム解釈実行許可入力手段と、

からなり、

解釈実行許可情報に従って、プログラム解釈実行禁止手段は、プログラム解釈実行手段がプログラムを解釈実行するのを禁止することを特徴とする、プログラム実行装置。

【請求項3】 外部伝送路を通して送信される、プログラムと外部伝送路を通して送信される、プログラムの発信元の識別情報とを入力とする、プログラム実行装置であり、

プログラム実行装置は、

前記プログラム解釈実行手段と、

プログラムを解釈実行する第2の手段である、プログラム解釈実行副手段と、

前記プログラム発信先判定手段と、

プログラム解釈実行選択手段と、

からなり、

プログラム発信先判定手段の判定に従ってプログラム解釈実行選択手段が、プログラムがプログラム解釈実行手段で実行されるのを許可するかあるいは禁止するかを選択して、プログラムがプログラム解釈実行手段で実行されるのを禁止する場合には、プログラムをプログラム解

釈実行手段副手段で実行させることを特徴とする、プログラム実行装置。

【請求項4】 外部伝送路を通して送信される、プログラムと外部伝送路を通して送信される、プログラムの発信元の識別情報とを入力とする、プログラム実行装置であり、

プログラム実行装置は、

前記プログラム解釈実行手段と、

前記プログラム解釈実行副手段と、

前記プログラム発信先判定手段と、

プログラムの解釈実行を許可する、あるいは、禁止する、のいずれかを示す、前記解釈実行許可情報を、ユーザが入力する手段である、前記プログラム解釈実行許可入力手段と、

プログラム解釈実行禁止選択手段と、

からなり、

プログラム発信先判定手段の判定結果が是である場合には、プログラム解釈実行禁止選択手段は、プログラム解釈実行手段がプログラムを解釈実行することを許可し、

20 プログラム発信先判定手段の判定結果が非である場合には、プログラム解釈実行禁止選択手段は、プログラム解釈実行手段がプログラムを解釈実行することを禁止することを特徴とし、

プログラム発信先判定手段の判定結果が非である場合には、プログラム解釈実行禁止選択手段は前記解釈実行許可情報を従って、プログラム解釈実行副手段がプログラムを解釈実行することを許可する、あるいは禁止する、のいずれかを選択することを特徴とする、プログラム実行装置。

30 【請求項5】 プログラム発信先判定手段の判定結果が否である場合、プログラムの解釈実行を許可するか否かをユーザに問い合わせる手段である、プログラム解釈実行許可確認手段を備えることを特徴とした、請求項2および請求項4のプログラム実行装置。

【請求項6】 識別情報はIPアドレスを含み、プログラム発信先判定手段は、IPアドレスを用いて判定を行うことを特徴とする、請求項1、2、3、4、5記載のプログラム実行装置。

40 【請求項7】 識別情報はURLを含み、プログラム発信先判定手段105は、URLを用いて判定を行うことを特徴とする、請求項1、2、3、4、5記載のプログラム実行装置。

【請求項8】 識別情報は暗号化された署名を含み、プログラム発信先判定手段は、暗号化された署名を用いて判定を行うことを特徴とする、請求項1、2、3、4、5記載のプログラム実行装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明はネットワーク通信によりプログラムをダウンロードし、プログラムを解釈実

行する環境、特に家電機器など、使用目的が特化されたプログラムを解釈実行する環境において、ダウンロードされたプログラムの検証を行なう装置およびその方法に関する。

【0002】

【從來の技術】近年、家電機器に対するプログラム配信のためのネットワーク機能搭載の要求が高まっている。

【0003】ネットワークを通じてプログラムの配信が行われることで、例えば、家電機器に対する機能の更新や追加が容易になり、新たなサービスの提供が可能になる、などの多くの利点が生まれる。

【0004】既にコンピューティングの分野では、こういったプログラムの配信は頻繁に行なわれている。

【0005】その代表的なものに、インターネットのウェブサーバと、ブラウザを載せたPCの間のネットワークがある。このネットワークにおいては、ネットワークに接続する者であれば誰でも、そこに流通するコンテンツを自由にダウンロードして利用できる。ここではこの性質を持つネットワークを「開かれたネットワーク」と呼ぶことにする。

【0006】開かれたネットワークの元では、利用者はネットワークに存在するプログラムもまた、自由にダウンロードし、実行することができる。

【0007】しかし一方で、開かれたネットワークでは、以下のことをネットワーク利用者によるリスクの分担の上で行なうという特徴を持つ。1. 接続先が意図した相手であること（認証と許可）。2. コンテンツが伝送経路で改ざんや盗聴がなされないこと（完全性と秘匿性）。3. コンテンツがそれを利用する環境で危険な振る舞いをしないこと（安全性）。

【0008】つまり開かれたネットワークとは、上記の1. 2. 3. が完全に保証される環境ではない。そのため、悪意ある第3者によって作成された危険なプログラムを、そうとは知らずに利用者がダウンロードし、実行してしまう可能性がある。

【0009】一方、開かれたネットワークとは異なり、流通するコンテンツをダウンロードする際に、コンテンツの暗号化やコンテンツをダウンロードする際の利用者の認証情報の確認などの必要を課すことによって、上記の1. 2. 3. の点を保証するネットワークも存在する。この性質を持つネットワークを「閉じられたネットワーク」と呼ぶことにする。

【0010】家電機器には一般的に高い信頼性・安全性が求められており、家電機器に対するプログラム配信の際には、信頼できる配信元が作成した安全なプログラムのみがダウンロードされることが求められる。

【0011】加えて家電機器に対するプログラム配信には、しばしば課金やデータの保護などが安全・正確に行われることが必要である。これを実現するには開かれたネットワークでは不充分である。これらの観点から、家

電機器に対するコンテンツの配信は、閉じられたネットワークが使用されることが望ましい。

【0012】実際、現行の家電機器向けを想定したネットワークは、何らかの手段によって上記の1. 2. 3. の点を保証した、閉じられたネットワークであるものが殆どである。

【0013】例えば現行の電話を用いた各種のサービスも、電話のネットワークを利用し、通話によるネットワークの利用に伴う課金を正確に行なう必然性から、そのサービスは閉じられたネットワークで実施されている。もちろん、コンテンツに暗号化等の処理を加えることにより、より強固で安全なネットワークとすることも可能である。

【0014】

【発明が解決しようとする課題】しかしながら近年、家電機器分野においても、インターネットに接続可能な携帯電話などのサービスに代表されるように、開かれたネットワーク環境に接続することが、一部の機器では可能になりつつある。インターネットに代表される開かれたネットワークと、家電機器との親和性は、今後さらに高まっていくであろうことが予想される。

【0015】このような状況においては先に述べたように、閉じられたネットワークを用い、信頼できる発信元からのみ、保証されたプログラムをダウンロードし実行することを、家電機器に遵守させることは難しくなる。

【0016】一つの解決策は、プログラムを解釈実行する実行環境のほうに、プログラムが危険なものでないかどうかを検査する検証機構を持たせる、というものである。

【0017】この解決策をとっている代表的なアーキテクチャにはJavaがある。

【0018】開発言語としてのJava言語は、オブジェクト指向言語としての開発効率の高さ、バグが出難い言語仕様、移植性の容易さ、既存の開発環境の豊富さ、など多くの優れた点を持っている。

【0019】さらにJavaアーキテクチャは、Java仮想マシンというインタプリタ機構によりプログラムを解釈実行する、という仕組みを持ち、機器のハードウェアやOSに依存しないプログラムを開発できるため、家電分野においてプログラム配信を実現するのに有望なアーキテクチャとみなされている。

【0020】Javaは、ネットワークを用いたプログラム（コンテンツ）配信の際に、セキュリティに関して高い安全性を保持していることでも知られている。具体的にはJavaは、以下のような検証機構を持っている。

【0021】(Laura Lemay, Charles L. Perkins, "Teach Yourself Java in 21 days" Prentice Hall, 1996) 1. Java言語そのものが安全性の高い言

語仕様を持っており、Java言語によって悪意ある変造コードを作成することは困難になっている。例えば、メモリに対する参照値などを、プログラマは直接操作することはできない。2. Java仮想マシンは、実行形式（Javaクラスファイル）が正しい規則に沿って記録されているかどうかを検証する機構を備える。これにより、変造された実行形式を実行解釈することを、Java仮想マシンは拒絶する。3. 仮想マシンのプログラム・ローダは記憶範囲をグループ別に管理しており、「保護の弱い」記憶範囲に記録されたクラスファイル（実行形式）が、「より保護の強い」記憶範囲に記録されたクラスファイルを侵害したり、許可なく参照したりすることはできない。例えば、インターネットから配信されたクラスファイルが、ローカルな領域に記録されていたクラスファイルに、許可なく置き換えられたりすることはない。4. Java APIが提供する機能群は、プログラムが破壊的な動作を行なえないように設計されている。

【0022】以上のようにJavaは、開発言語からプログラム実行系までの各レベルにおいて、強固な検証機構を有しているが、このような機能を持たせたため、Java仮想マシンはその動作のために多大な使用メモリ量と高性能のプロセッサを必要としている。

【0023】例えば2を実現するために、仮想マシンは実行形式中のJavaの命令コードに対してフロー解析（A. Aho, R. Sethi, J. Ullman, "compiler Principles, Techniques, and Tools", Addison-Wesley, 1986）を行わなければならず、これはJava仮想マシンの実装コードサイズと使用メモリ量の増大を招く（試算によると、2. の検証機構を備えない場合の総使用メモリ量と比較して、1.5～3.0倍増大する）。

【0024】さらにこれらの検証処理は、プログラムの実行中にもしばしば発生するものであるため、Java仮想マシンの処理の負荷を大きくし、高速にプログラムを動作させることへの妨げにもなる。

【0025】家電機器分野ではコストに対する要求は非常に厳しく、コンピューティング分野以上に省資源・低成本・低消費電力が強く要求されるものが大多数である。

【0026】例えば家電機器にJavaを搭載させるアプローチの代表的なものに、サンマイクロシステムズが家電組み込み向けに作成した仕様であるEmbedded Javaアーキテクチャがあるが、プログラムの動作のためには最低、RAM 512Kバイト、ROM 512Kバイト、25MHz以上のCPUを必要としている（Deepak Mutchandani, "Java for Embedded Systems", IEEE INTERNET COMPUTING, May/June 19

98）。これはハイエンドの情報家電機器に対して搭載可能、というレベルを実現したに留まるものであり、必要メモリ量は依然として一般的な家電機器にとって大きなサイズである。

【0027】特に、携帯電話や白物家電などには、低動作周波数の8/16bitマイコンや、数十KB程度のメモリしか搭載しないものが主流であり、このような機器に対してはEmbedded Javaを適用することは不可能である。

10 【0028】以上に示される通り、Javaに代表されるような、検証機構を実行環境に搭載してネットワークを流れるプログラムの安全性を保証するというアプローチは、一般的な家電機器全般に適用するには非常に困難である。

【0029】本発明は、プログラム配信のためのネットワーク機能を備えた家電機器における問題である、ダウンロードしたプログラムの安全性・秘匿性・完全性の保証を、可能な限り低コストで簡便に実現するための手段を提供するものである。

20 【0030】

【課題を解決するための手段】本発明の装置は、外部伝送路を通して、プログラムと、プログラムの発信元の識別情報を受信する。装置は受信した識別情報を判定し、その結果により、プログラムが安全なものであるか、それとも安全性の保障できない、実行環境とユーザに危険を及ぼす可能性のあるものであるか、を判断する。

【0031】安全と判断したプログラムに対しては、解釈実行を許可する。この解釈実行の手段はプログラムに対して特別に検証機構による処理を必要としないものを想定している。

【0032】よってプログラムを、使用メモリ量を少なく抑えたまま、高速に動作させることができる。

【0033】安全性が保証できないプログラムに対しては、本発明は以下のようないくつかの対処方法を提供している。

【0034】1) 安全性が保証できないプログラムは一切解釈実行させない。

【0035】この方法を取る場合、プログラム実行装置は危険なプログラムを一切動作させないため、検証機構を備える必要はない。プログラムの解釈実行のために必要な資源量は小さく抑えたまま、安全と判断されたプログラムは全て高速に動作させることができる。

【0036】2) ユーザによって与えられた、プログラムの解釈実行を許可する、あるいは禁止する、のいずれかを示す情報を元に、プログラムの解釈実行を決定する。

【0037】この方法を取る場合、プログラム実行装置は安全性が保証できないプログラムを動作させるかどうかをユーザの判断に任せる。

【0038】本来家電機器はその特徴上、信頼できる特定の（おそらくは機器を製造したメーカーの保証を受け

た) 発信元からのみ、プログラムをダウンロードし実行することを推奨するものである。そのような背景においても、ユーザが開かれたネットワークから、プログラム(コンテンツ)のダウンロードを行うというのは、例外的な状況下においてあると考えられる。

【0039】つまり、ユーザにとって何らかの理由によって有益と思われるプログラム・コンテンツを、その機器を使用して獲得したい場合である。

【0040】開かれたネットワークから、プログラム(コンテンツ)を獲得する場合、ユーザはそのダウンロード対象プログラム・コンテンツが安全なものであるとの知識を持っていることが前提であり、ユーザは自己の責任においてその行為を行わなければならない。

【0041】2) の方法は、そのユーザの判断を、プログラム実行装置に反映させる方法である。

【0042】プログラムはユーザの自己責任によってプログラムを解釈実行させるため、検証処理は行われないことが前提となり、プログラムの解釈実行のために必要な資源量は小さく抑えたまま、プログラムを高速に動作させることができる。

【0043】3) ダウンロードしたプログラムが危険なものである可能性があることをユーザに知らせた後、2) を行う。

【0044】2) と同じであるがプログラムを動作させる前に、判断を促すための通告をユーザに対して行うことができる。

【0045】ユーザはそれを用い、より正確な判断で、プログラムの解釈実行の許可や禁止の情報を装置に対して与えることができる。

【0046】4) プログラム実行装置に、プログラムの解釈実行のための、従来とは別の第2の手段を用意し、安全性が保証できないプログラムはその第2の手段により実行することにする。

【0047】安全性が保障されるプログラムとされないプログラムの解釈実行のための手段を分離する方法である。

【0048】この場合、安全性が保証されないプログラムを解釈実行する手段は、検証処理を動作させつつプログラムを解釈実行すること、あるいは、プログラム実行装置の他の手段には影響を与えることなく独立して存在していることが望ましい。

【0049】これにより、安全なプログラムを高速に動作させることを保証したまま、危険なプログラムもまた動作させることができる。5) 2) と4) の方式を同時にとるもの6) 3) と4) の方式を同時にとるもの本発明はこれらの手段を、プログラムを実行する機器に対して提供するものである。

【0050】

【発明の実施の形態】本発明は、外部伝送路に接続されたプログラム実行環境全般に適用可能なものであるが、

特に省資源・低コストが要求される家電機器への適用を強く意図したものである。

【0051】ここでは特に、ディスプレイを装備したインターネットに接続可能な携帯端末に対して、本発明を適用した例を適宜加えながら、説明を行っていく。

【0052】以下、本発明の実施の形態について、図面を用いて説明する。

【0053】まず図1は、請求項1に記載のプログラム実行装置の構成を示したものである。

10 【0054】図1の装置の処理のフローチャートを図5に示す。

【0055】これらについて順に説明していく。

【0056】外部伝送路101から受信する、プログラム102と識別情報103は、例えばインターネットなどのネットワークを通じてダウンロードする。

【0057】プログラムは、装置が備える実行系により解釈実行が可能な形式を持ったデータとして獲得することができる。例えば装置がJava仮想マシンを備えているならば、その装置はJavaの実行形式であるJava

20 クラスファイルが実行可能であるため、Javaクラスファイルを外部伝送路を通じてダウンロードする。

【0058】識別情報とは、プログラムをダウンロードする時に、例えば、その発信元が特定されるものであるとする。本実施例では、説明を簡単にするために、これをプログラムの名前で代用する。もちろん他の情報であってもよい。

【0059】識別情報として用いることが可能なものに、例えば、インターネットに接続した機器に固有に割り振られるIPアドレスや、Web, FTP, Gopher, News, TELNETなどの情報を指定する世界共通で一意なアドレスであるURL (Uniform Resource Locator) がある。

【0060】米国Netscape Communications CorporationによるSSL (Secure Sockets Layer) プロトコルでは、相手の認証や使用する暗号やデジタル署名のアルゴリズムなどに関するネゴシエーションを行ない、相互に認証してから、データの送受信を行っているが、もちろんSSLに使用されるものに代表される暗号化された署名を識別情報とすることも可能である。もちろん、識別情報は上記に挙げたもの以外のものであってもよい。

【0061】図5において最初の処理であるP105では、識別情報103を解読し、識別情報が、信頼されたプログラム発信元を示しているか否かを判定する。

【0062】これは図1における、プログラム発信先判定手段105によって行われる処理である。

【0063】この処理P105の具現は、何らかの形で識別情報を解釈して、その信頼性を判定する必要がある。例えば以下のような形で、これを実現することができる。

【0064】プログラム発信先判定手段105は、信頼できる発信元を示した全ての識別情報を記録したデータを保持している。信頼できるか否かの判定は、ダウンロードしてきた識別情報をこのデータと照合し、識別情報がデータに登録されているなら、この識別情報は信頼できると判定し、登録されていないなら信頼できないと判定する。

【0065】識別情報として、SSLプロトコルで用いられるような暗号化された署名を用いている場合には、プログラム発信先判定手段105は、暗号の解読に成功するか否かによって、識別情報を信頼できるかできないかを決定することもできる。もちろん、信頼性の判定は他の手段で行うこともできる。

【0066】処理P104は、プログラム102を解釈実行する。

【0067】これは図1における、プログラム解釈実行手段104によって行われる処理である。

【0068】本発明の備えるプログラム解釈実行手段104は、プログラムを解釈実行することが可能な環境が備える、従来の実行系と同じ仕組みであり、従来と同様にプログラムの実行解釈を行う。

【0069】処理P106は、プログラム102がプログラム解釈実行手段104によって解釈実行されるのを禁止する。

【0070】これは図1における、プログラム解釈実行禁止手段106によって行われる処理である。

【0071】プログラム解釈実行禁止手段106の実現には、例えば以下のようなものが考えられる。

【0072】プログラム解釈実行禁止手段106は内部に、図9に示されるテーブルを保持する。

【0073】このテーブルのエントリは、ダウンロードしたプログラムの識別子と、それに対応する、プログラムの状態を示す値とで構成される。プログラムの状態を示す値は、permittedもしくはforbiddenという2値のどちらかを取る。テーブルに記録される時の初期状態はpermittedであるとする。

【0074】プログラム解釈実行手段104は、解釈実行を始める前に図9のテーブルを参照して、解釈実行しようとするプログラムの識別子を検索する。該当するプログラムの識別子のエントリの、プログラムの状態を示す値がforbiddenであるか、あるいはエントリがテーブルにない場合には、そのプログラムの解釈実行を開始しないことにする。

【0075】この場合、処理P106は、該当するプログラムの識別子のエントリの、プログラムの状態を示す値をforbiddenに変更することによって実現できる。

【0076】ここではプログラム解釈実行禁止手段106の実現にテーブルを用いて実施例を示したが、もちろんテーブル以外の構造、例えばリスト、木、ハッシュを

用いて実現することも可能である。

【0077】処理P104、あるいは処理P106が終了すると、図1で行われる処理全体が完了する。

【0078】図2は、請求項5記載のプログラム実行装置の構成を示したものである。

【0079】図2から、プログラム解釈実行許可確認手段107を除いた図面が、請求項2記載のプログラム実行装置の構成を示すものになる。

【0080】図2の装置の処理のフローチャートを図6に示す。

【0081】図6において、P207の処理を「何もない」に置き換えたものが、請求項2記載の装置の処理のフローチャートになる。

【0082】これらについて順に説明していく。

【0083】外部伝送路101から受信する、プログラム102および識別情報103は、先に説明したものと同じである。

【0084】図6において最初の処理であるP205は、図2におけるプログラム発信先判定手段105によって行われる処理であり、先に説明したP105と同じ処理を行う。

【0085】処理P207は、プログラムの実行を許可するか否かをユーザに問い合わせる。

【0086】これは図2における、プログラム解釈実行許可確認手段107によって行われる処理である。

【0087】例えばこれは、装置が備えるディスプレイに、図10のような表示を行うことで実現できる。

【0088】処理P209は、プログラムの実行を許可するか否かを示す情報である、プログラム解釈実行許可情報108を得る。

【0089】これは図2における、プログラム解釈実行許可入力手段109によって行われる処理である。

【0090】装置のユーザは、処理P209が行われる前に、何らかの意思表明を装置に対して行う。プログラム解釈実行許可入力手段は、その意思表明を受け、装置内部のデータとして、プログラム解釈実行許可情報108を生成する。

【0091】ユーザによる意思表明は、装置が備える入力装置、例えばボタン、スイッチ、ダイヤル、マウス、トラックボール、ジョイスティック、等による入力操作の組み合わせによって、あるいはこれらの操作を何もないことによってなされる。

【0092】例えば、ユーザは、"機能"ボタンを押した後、"0"ボタンを押す、という行為を、装置に対して行う。これによって装置は、全てのプログラムの実行が許可されたという情報を得る。・ユーザは、ダイヤルによって"m a z e. c j"という文字列をディスプレイ上で選択した後に、"1"ボタンを押す。これによって装置は、"m a z e. c j"という識別名を持つプログラムの実行が禁止されたという情報を得る。・ユーザ

は、処理P209が行われる前に、プログラムの実行の許可／禁止を指示する特定の入力を、何ら装置に対して与えなかつたとする。これによって装置は、"m a z e. c j" という識別名を持つプログラムの実行が禁止されたという情報を得る。

【0093】ユーザの意思表明は、処理P207と連携して実施されても良い。例えば、・図10の表示が行なわれた後に、"0"ボタンを押す、という行為を、装置に対して行う。これによって装置は、"m a z e. c j" という識別名を持つプログラムの実行が許可されたという情報を得る。

【0094】ここでは"0", "1", "機能"というボタンとダイヤル操作の組み合わせによる実施例を示したが、もちろんボタンの種類や押す順番は上に挙げたもの以外であつても良いし、入力装置の組み合わせも他のものであつても良い。

【0095】処理P206Aは、プログラム解釈実行許可情報108を解釈して、ユーザがプログラム102の実行を許可したか否かを判定する。

【0096】これは図2において、プログラム解釈実行禁止手段106によって行われる処理である。

【0097】処理P204は、プログラム102を解釈実行する。

【0098】これは図2における、プログラム解釈実行手段104によって行われる処理であり、先に説明した処理P104と同じものである。

【0099】処理P206Bは、プログラム102の解釈実行を禁止する。

【0100】この処理は、図2のプログラム解釈実行禁止手段106によって行われる処理であり、先に説明した処理P106と同じものである。

【0101】処理P204、あるいは処理P206Bが終了すると、図2で行われる処理全体が完了する。

【0102】図3は、請求項3に記載のプログラム実行装置の構成を示したものである。

【0103】図3の装置の処理のフローチャートを図7に示す。

【0104】これらについて順に説明していく。

【0105】外部伝送路101から受信する、プログラム102および識別情報103は、先に説明したものと同じである。

【0106】図7において最初の処理であるP305は、図3におけるプログラム発信先判定手段105によって行われる処理であり、先に説明したP105と同じ処理を行う。

【0107】処理P304は、プログラム102をプログラム解釈実行手段104によって実行する。

【0108】これは図3における、プログラム解釈実行選択手段111と、プログラム解釈実行手段104によって行われる処理である。

【0109】処理P310は、プログラム102をプログラム解釈実行副手段110によって実行する。

【0110】これは図3における、プログラム解釈実行選択手段111と、プログラム解釈実行副手段110によって行われる処理である。

【0111】プログラム解釈実行副手段110は、プログラムを解釈実行することが可能な環境が備える、従来の実行系と同じ仕組みであり、従来と同様にプログラムの実行解釈を行うが、これはプログラム解釈実行手段104とは別の手段として装置内部に実現される。

【0112】プログラム解釈実行選択手段111の実現には、例えば以下のようないが考えられる。

【0113】プログラム解釈実行選択手段111は内部に、図11に示されるテーブルを保持する。

【0114】このテーブルのエントリは、ダウンロードしたプログラムの識別子と、それに対応する、プログラムの状態を示す値とで構成される。

【0115】プログラムの状態を示す値は、securedもしくはunsecuredという2値のどちらかを取る。

【0116】プログラムの実行の前に、プログラム解釈実行選択手段111は、プログラム発信先判定手段105の判定結果によって、信頼された発信元からのプログラムであればsecuredを、そうでなければunsecuredを記録する。

【0117】プログラム解釈実行手段104は、解釈実行を始める前に図11のテーブルを参照して、解釈実行しようとするプログラムの識別子を検索する。該当するプログラムの識別子のエントリの、プログラムの状態を示す値がsecuredである場合にのみ、そのプログラムの解釈実行を開始する。

【0118】プログラムの状態を示す値がunsecuredであるか、あるいはエントリがテーブルにない場合には、そのプログラムの解釈実行はプログラム解釈実行副手段110によって実行される。

【0119】ここではプログラム解釈実行選択手段111の実現にテーブルを用いて実施例を示したが、もちろんテーブル以外の構造、例えばリスト、木、ハッシュを用いて実現することも可能である。

【0120】処理P304、あるいは処理P310が終了すると、図3で行われる処理全体が完了する。

【0121】図4は、請求項5記載のプログラム実行装置の構成を示したものである。

【0122】図4から、プログラム解釈実行許可確認手段107を除いた図面が、請求項4記載のプログラム実行装置の構成を示すものになる。

【0123】図4の装置の処理のフローチャートを図8に示す。

【0124】図8において、P407の処理を「何もない」に置き換えたものが、請求項4記載の装置の処理

のフローチャートになる。

【0125】これらについて順に説明していく。

【0126】外部伝送路101から受信する、プログラム102および識別情報103の説明は、先に説明したものと同じである。

【0127】図8において最初の処理であるP405は、図4におけるプログラム発信先判定手段105によって行われる処理であり、先に説明したP105と同じ処理を行う。

【0128】処理P407は、プログラムの実行を許可するか否かをユーザに問い合わせる。

【0129】これは図4における、プログラム解釈実行許可確認手段107によって行われる処理であり、先に説明したP207と同じ処理を行う。

【0130】処理P409は、プログラムの実行を許可するか否かを示す情報である、プログラム解釈実行許可情報108を得る。

【0131】これは図4における、プログラム解釈実行許可入力手段109によって行われる処理であり、先に説明したP209と同じ処理を行う。

【0132】処理P406Aは、プログラム解釈実行許可情報108を解釈して、ユーザがプログラム102の実行を許可したか否かを判定する。

【0133】これは図4において、プログラム解釈実行禁止手段106によって行われる処理である。

【0134】処理P404は、プログラム102をプログラム解釈実行手段104によって実行する。

【0135】これは図4における、プログラム解釈実行禁止選択手段112と、プログラム解釈実行手段104によって行われる処理である。

【0136】処理P410は、プログラム102をプログラム解釈実行副手段110によって実行する。

【0137】これは図4における、プログラム解釈実行禁止選択手段112と、プログラム解釈実行副手段110によって行われる処理である。

【0138】処理P406Bは、プログラム102の解釈実行を禁止する。

【0139】この処理は、図4のプログラム解釈実行禁止選択手段112によって行われる処理である。

【0140】プログラム解釈実行禁止選択手段112の実現には、例えば以下のようなものが考えられる。

【0141】プログラム解釈実行禁止選択手段112は内部に、図12に示されるテーブルを保持する。このテーブルのエントリは、ダウンロードしたプログラムの識別子と、それに対応する、プログラムの状態を示す値とで構成される。

【0142】プログラムの状態を示す値は、`sucurred`もしくは`permitted`もしくは`forbidden`という3つの値のいずれかを取る。ここではエントリに最初に記録されるときの初期値は`secured`

であるとする。

【0143】処理P406Aにおいて、プログラム解釈実行選択手段111は、プログラム解釈実行許可情報108を解釈して、ユーザが実行を許可していれば、プログラムの状態を示す値に`permitted`を、そうでなければ`forbidden`を記録する。

【0144】プログラム解釈実行手段104は、プログラム102の解釈実行を始める前に図11のテーブルを参照して、解釈実行しようとするプログラムの識別子を検索する。該当するプログラムの識別子のエントリの、プログラムの状態を示す値が`secured`である場合にのみ、そのプログラムの解釈実行を開始する。

【0145】プログラムの状態を示す値が`permitted`である場合には、プログラム102はプログラム解釈実行副手段110によって解釈実行される。

【0146】プログラムの状態を示す値が`forbidden`であるか、あるいはエントリがテーブルにない場合には、プログラム102は解釈実行されない。

【0147】ここではプログラム解釈実行禁止選択手段20 112の実現にテーブルを用いて実施例を示したが、もちろんテーブル以外の構造、例えばリスト、木、ハッシュを用いて実現することも可能である。

【0148】処理P404、あるいは処理P410、あるいは処理P406Bが終了すると、図4で行われる処理全体が完了する。

【0149】以上が本発明の実施の形態である。

【0150】

【発明の効果】以上のように本発明は、外部伝送路を通して、プログラムと、プログラムの発信元の識別情報を受信し、その識別情報が、信頼できる発信元を示しているかどうかを判定する。

【0151】判定の結果、信頼できる発信元のものであることが判別できない場合、実行環境は、そのことをユーザ（実行環境の使用者）に知らせ、ユーザによって与えられた情報を元に、プログラムの解釈実行を決定する。もしくは、従来の解釈実行手段とは異なる、第2の解釈実行手段によって、そのプログラムを解釈実行する。

【0152】このような機構を設けることにより、プログラムを実行する装置は、安全であると保証されるプログラムは従来通り、使用資源が少なくかつ高速な解釈実行を行うことができる。かつダウンロードしたプログラムの安全性・秘匿性・完全性の保証を、装置の低コスト・省資源性を保ったまま行うことができる。

【図面の簡単な説明】

【図1】本発明のプログラム実行装置の構成例を示す図

【図2】本発明のプログラム実行装置の構成例を示す図

【図3】本発明のプログラム実行装置の構成例を示す図

【図4】本発明のプログラム実行装置の構成例を示す図

【図5】本発明のプログラム実行装置の動作例を示すフ

ローチャート

【図6】本発明のプログラム実行装置の動作例を示すフローチャート

【図7】本発明のプログラム実行装置の動作例を示すフローチャート

【図8】本発明のプログラム実行装置の動作例を示すフローチャート

【図9】プログラム解釈実行禁止手段の構成例を示す図

【図10】プログラム解釈実行許可確認手段による表示

例を示す図

【図11】プログラム解釈実行選択手段の構成例を示す*

*図

【図12】プログラム解釈実行禁止選択手段の構成例を示す図

【符号の説明】

101 通信伝送路

102 プログラム

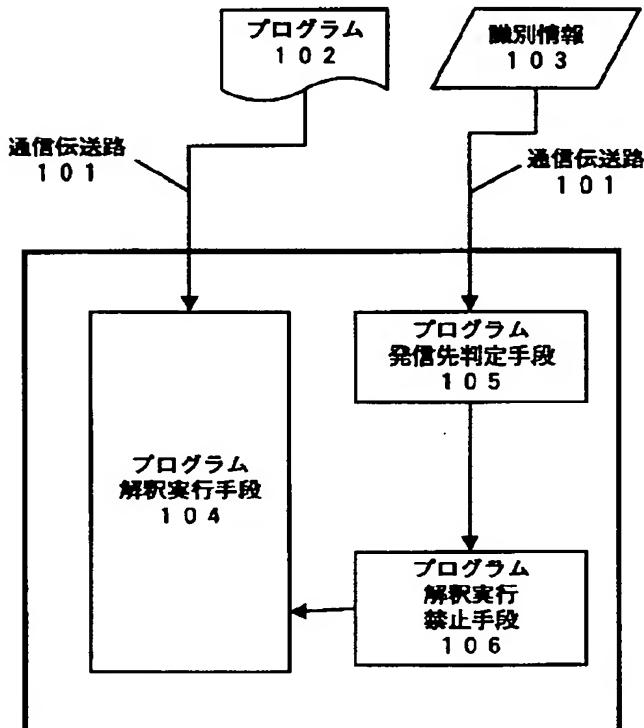
103 識別情報

104 プログラム解釈実行手段

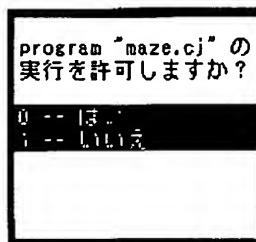
105 プログラム発信先判定手段

10 106 プログラム解釈実行禁止手段

【図1】



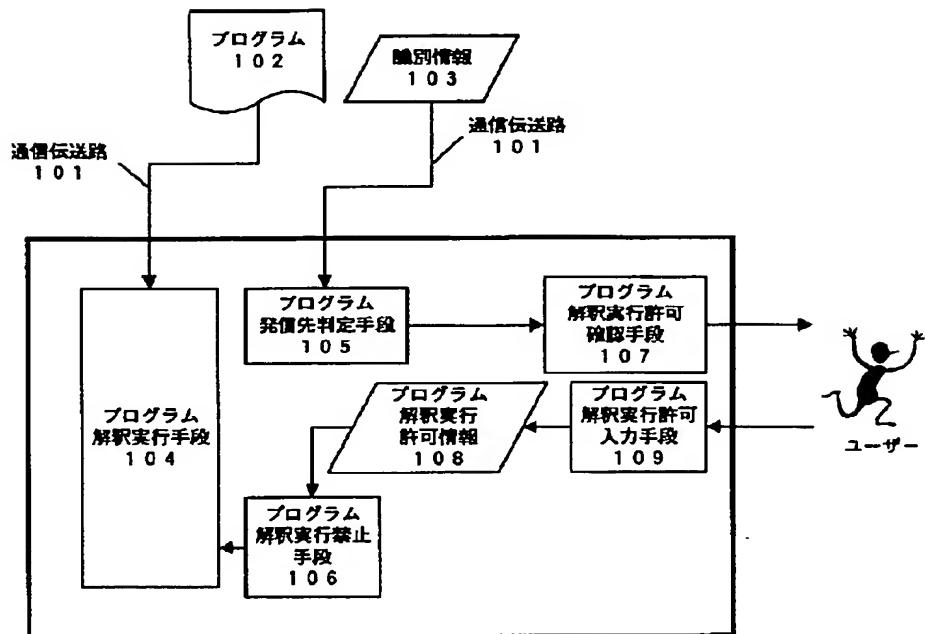
【図10】



プログラム解釈実行許可確認手段の実施例

プログラム実行装置・その1

【図2】



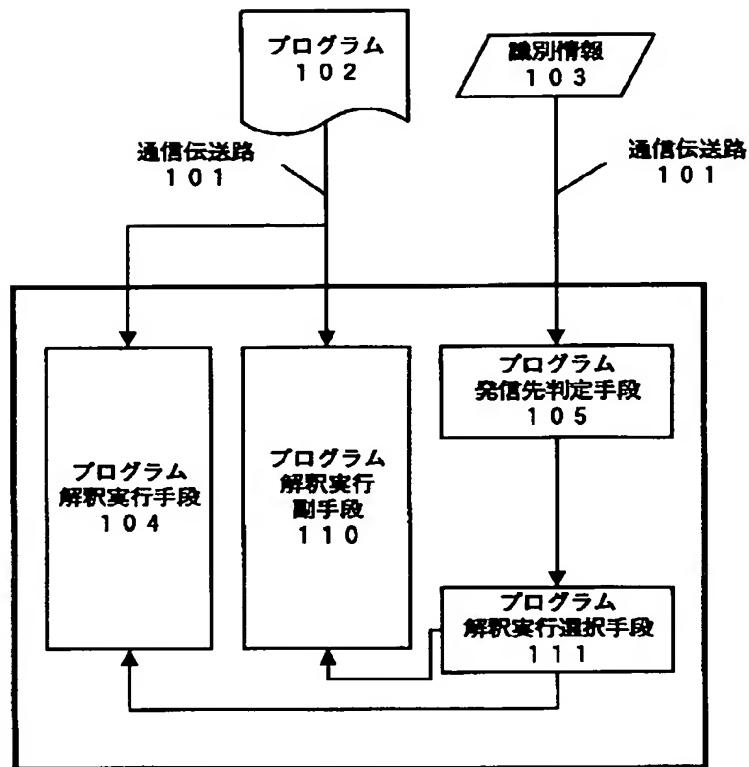
プログラム実行装置・その2

【図9】

| プログラムの識別子 | プログラムの状態 |
|------------------|-----------|
| ButtonCrusher.c] | forbidden |
| Maze.c] | permitted |
| Clipboard.c] | permitted |
| AddressWriter.c] | forbidden |
| DigitalClock.c] | permitted |
| ⋮ | ⋮ |

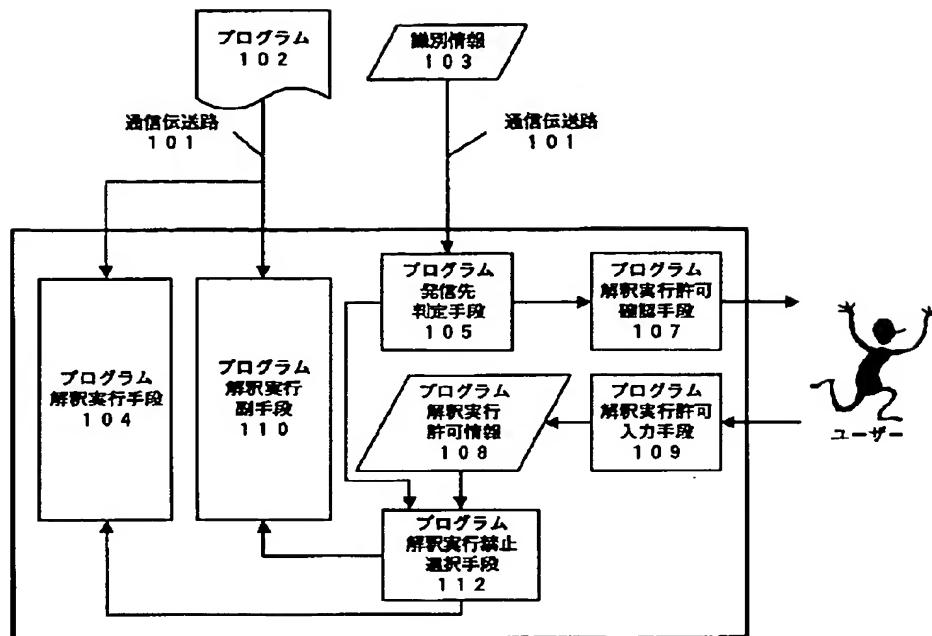
プログラム解釈実行禁止手段の実施例

【図3】



プログラム実行装置・その3

【図4】



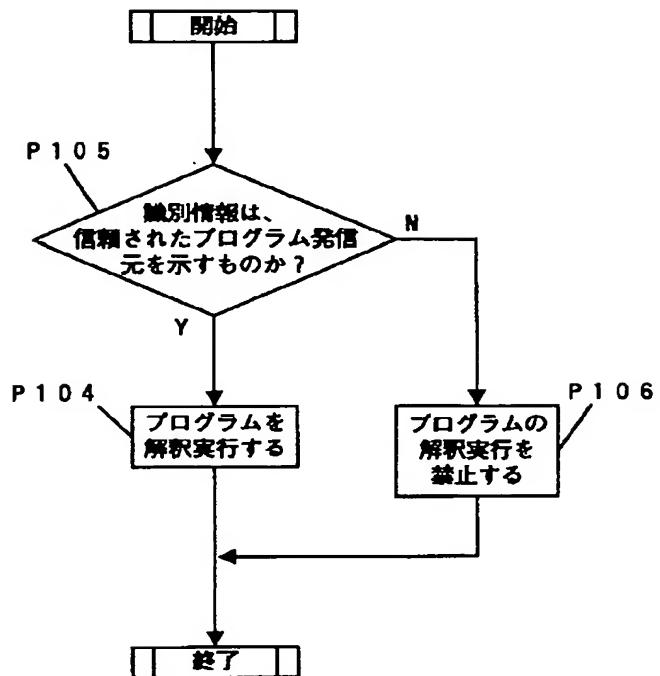
プログラム実行装置・その4

【図11】

| プログラムの識別子 | プログラムの状態 |
|------------------|-----------|
| ButtonCrusher.cj | secured |
| Maze.cj | unsecured |
| ClipBoard.cj | unsecured |
| AddressWriter.cj | secured |
| DigitalClock.cj | unsecured |
| ⋮ | ⋮ |

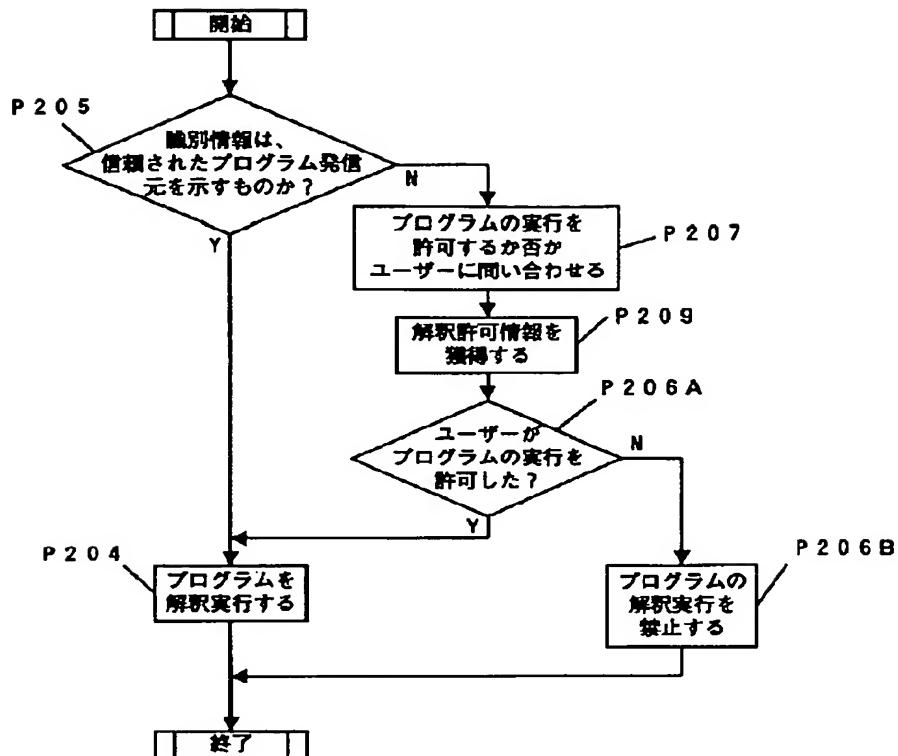
プログラム解釈実行選択手段の実施例

[図5]



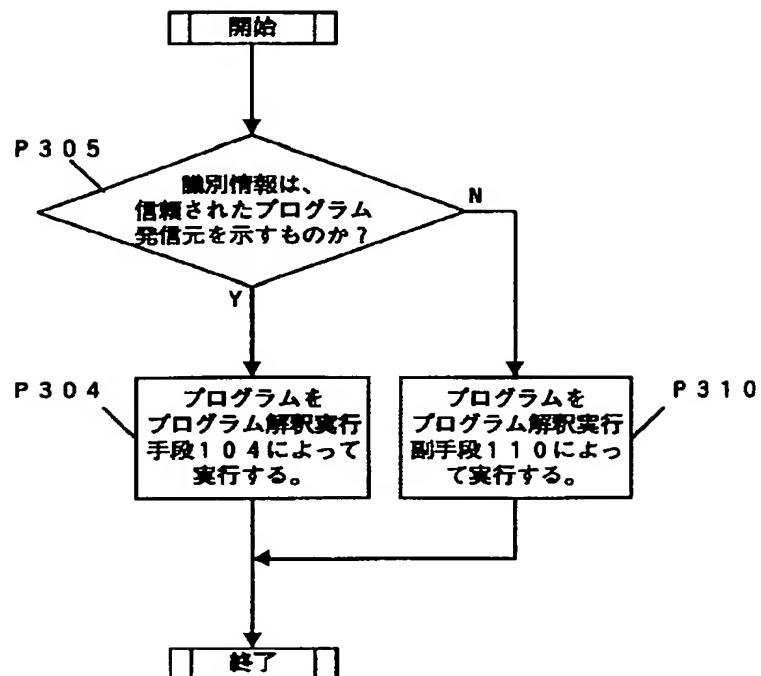
プログラム実行方式の処理手順・その1

〔図6〕



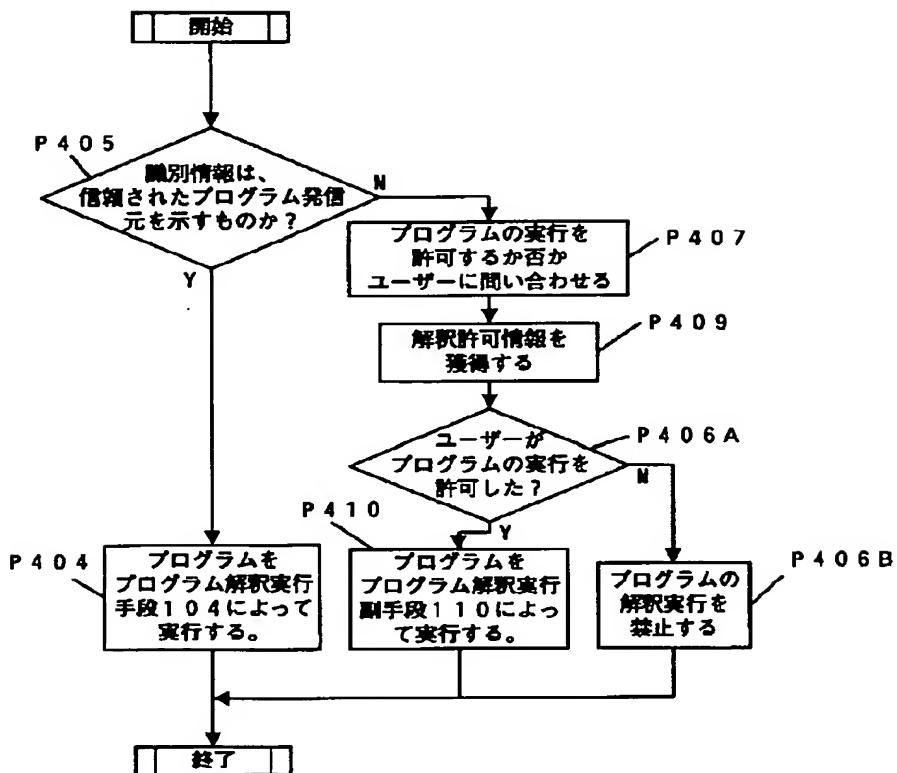
プログラム実行方式の処理手順・その2

〔図7〕



プログラム実行方式の処理手順・その3

【図8】



プログラム実行方式の処理手順・その4

【図12】

| プログラムの識別子 | プログラムの状態 |
|------------------|-----------|
| ButtonCrusher.cl | forbidden |
| Maze.cj | secured |
| ClipBoard.cj | permitted |
| AddressWriter.cj | forbidden |
| DigitalClock.cj | secured |
| ⋮ | ⋮ |

プログラム解釈実行禁止選択手段の実施例